Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of claims:

1. (currently amended) A method of an executable program of a cryptographic processor masking a conditional jump operation in said cryptographic processor, said cryptographic processor being programmed such that said executable program executes a sequence of instructions, wherein the conditional jump is determined by said executable program evaluating a distinguishing value V against a reference value and wherein the reference value is bounded by an upper limit Vmax and a lower limit Vmin, the method comprising the steps of:

(a) determining a location of said conditional jump in said executable program; and

(b) inserting processor instructions at said location to direct execution of said program to one of two branches, said processor instructions computing a target address in said processor instructions, the target address being derived from said distinguishing value and a base address constituted by a random number, wherein for each evaluation of said distinguishing value against said reference value one of said branches is selected and a ~~different~~ random number of processor instructions are executed [[for]] within said branch ~~each conditional jump~~ for directing program flow to said one of two branches.

2. (currently amended) A method as defined in claim 1, wherein said distinguishing value [[being]] is combined with said random number, thereby adding a random number of instructions on every conditional evaluation.

3. (currently amended) A method as defined in claim 1, wherein said inserted instructions ~~including~~ include calls to respective subroutines, said subroutines including instructions for changing the return address of the subroutines to said one of two branches.

4. (cancelled)

5. (currently amended) A method as defined in claim 1, wherein said target address is computed using an extended addressing mode of said processor.

6. – 26. (cancelled)

27. (new) A method of performing an executable program of a cryptographic processor to mask a conditional jump operation in said cryptographic processor, said cryptographic processor being programmed such that said executable program executes a sequence of instructions that includes a set processor instructions to direct execution of said program to one of two branches, wherein the conditional jump is determined by said executable program evaluating a distinguishing value V against a reference value and wherein the reference value is bounded by an upper limit Vmax and a lower limit Vmin, the method comprising the steps of computing at a location of said conditional jump in said executable program, a target address, the target address being derived from said distinguishing value and a base address constituted by a random number, wherein for each evaluation of said distinguishing value against said reference value one of said branches is selected and a random number of processor instructions are executed within said branch when directing program flow to said one of two branches.

28 (new). A method as defined in claim 27, wherein said distinguishing value is combined with said random number, thereby adding a random number of instructions on every conditional evaluation.

29. (new) A method as defined in claim 27, wherein said inserted instructions include calls to respective subroutines, said subroutines including instructions for changing the return address of the subroutines to said one of two branches.

30.(new) A method as defined in claim 27, wherein said target address is computed using an extended addressing mode of said processor.

31. (new) A cryptographic token having a cryptographic processor to perform a conditional jump operation in an executable program, said cryptographic processor being programmed such that said executable program executes a sequence of instructions that include a set of processor instructions, wherein the conditional jump is determined by said executable program evaluating a distinguishing value V against a reference value and wherein the reference value is bounded by

an upper limit Vmax and a lower limit Vmin, said executable program computing at a location of said conditional jump in said executable program, a target address in said processor instructions, the target address being derived from said distinguishing value and a base address constituted by a random number, wherein for each evaluation of said distinguishing value against said reference value one of said branches is selected and a random number of processor instructions are executed within said branch for directing program flow to said one of two branches.

32. (new) A token as defined in claim 31, wherein said distinguishing value is combined with said random number, thereby adding a random number of instructions on every conditional evaluation.

33. (new) A token as defined in claim 31, wherein said inserted instructions include calls to respective subroutines, said subroutines including instructions for changing the return address of the subroutines to said one of two branches.

34. (new) A token as defined in claim 31, wherein said target address is computed using an extended addressing mode of said processor.